Midterm I Exam, spring 2013 | **Saturday March 19th, 2013** | **Exam time: 07:00-9:00 P.M.**

Student's name:... **ID:** .... **Section:** ....

Problem 1

(a) Give the following functions a number in order of increasing asymptotic growth rate. If two functions have the same asymptotic growth rate, give them the same number.

Function	Rank
$8^{\lg n} + 2n$	
$5 \lg n^8$	
$n^3 + 5n^2 - 100$	
$\dfrac{n^2}{\lg n}$	
$2\ \lg n + \lg(\lg n^2)$	
3^{2n}	

(b) Using the definition of θ, find $g(n)$, **C**, and n_0 in the following:

$$4n^6 - 2n^2 + n \in \theta(g(n))$$

Problem 2 (6 points)

Consider the following recurrence relation:

$$\mathrm{T(n)} = 2\mathrm{T}\left(\frac{n}{2}\right) + 2\mathrm{n}.$$

(a) Solve this recurrence relation using recursive substitutions.

(b) Find g(n), where **T(n) ∈ O (g(n)).**

Hint: $\sum_{i=0}^{n-1} 2^i = 2^n - 1.$

Problem 3 (6 points)

Solve the following recurrences using the Master theorem by giving tight θ-notation bounds. Justify your answers.

(a) $\mathbf{T(n) = 8T(n/2) + 5}n^2$

(b) $\mathbf{T(n) = 9T(n/3) + 3}n^2$

(c) $T(n) = 7T(n/2) + n^3$

Problem 4

For each algorithm listed below, give a recurrence that describes its worst-case running time, and give its worst-case running time using O-notation.
You need **not** justify your answers.

(a) Merge sort

(b) Insertion sort

(c) Quicksort algorithm

(d) Binary Search

Problem 5 (4 points)

Consider a variation of MergeSort which divides the list of elements into two lists of size 1/4 and 3/4, recursively at each step, instead of dividing it into halves. The Merge procedure does not change.

(a) Give a recurrence relation for this algorithm
(b) Draw a recursion tree for the algorithm
(c) Using the recursion tree, explain how you can deduce the worst case upper bound.

Problem 6

Consider the problem below, and suggest **TWO** algorithm design techniques and give a high-level description of the **TWO** algorithms to solve it.

a- There are n closed boxes numbered from 1 to n and you are told that there are k balls in the first k boxes (one ball in each box), and all other boxes are empty. How to find the value of k?

Master Theorem:

If **T(*n*) = *a* T(*n/b*) + f(*n*)** then

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\log n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta\left(f(n)\right) & f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right) \text{AND} \\ & af(n/b) < cf(n) \text{ for large } n \end{cases} \quad \begin{matrix} \varepsilon > 0 \\ c < 1 \end{matrix}$$